Remote Control of Web 2.0-enabled Laboratories from Mobile Devices

Diego López-de-Ipiña, Javier García-Zubia and Pablo Orduña
Faculty of Engineering, University of Deusto, Apdo 1, 48080 Bilbao,SPAIN
{dipina, zubia}@eside.deusto.es, pablo@ordunya.com

Abstract

The design of WebLabs has traditionally been focused on the hardware rather than the software side. However, paying more importance to the software side can bring about important improvements: richer collaboration among group members, better scalability, securer access and more natural anywhere at anytime access to the remote lab. Regarding this latter issue, enabling a user to access a remote WebLab not only through standard internet-connected PCs but also by means of their mobile devices can be very interesting and potentially advantageous. The core of this paper is to discuss the transformation of DeustoWebLab into a Web 2.0-compatible application accessible from mobile devices.

1. Introduction

Nowadays, the use of WebLabs or Remote Labs in universities is extending widely. A WebLab is a combination of hardware and software which allows a student to complete lab assignments remotely, e.g. from his home, just as if she was present in a laboratory. The student remotely controls the devices in the laboratory through a web interface, and monitors the outputs resulted from her interaction through a WebCam, a data file, or some kind of virtual instruments.

Web 2.0 [1] is a new buzzword coined to refer to the substantial innovation progress achieved by Web applications and portals in the last two year. For first time since the bursting of the dot-com bubble in the fall of 2001 the web industry has gathered a lot of interest. This is justified by the emergence of much more dynamic, responsive and user friendly new web applications such as Google Maps or Flicr, backed by industry giants such as Google or Yahoo!.

Some visionaries and leading companies are considering moving the new Web 2.0 paradigm from the desktop to mobile devices, giving place to the term Mobility 2.0. This should result in the development of much more interactive and responsive browsing applications for mobile devices. A key characteristic of Web 2.0 applications is that, thanks to their adoption of Ajax technology, fragments rather than whole web pages are changed after user interaction. Consequently, the amount of data transferred between web server and client application is reduced dramatically, and, therefore, very suitable for the mobile domain.

The main subject of this paper is to describe how to transform a conventional WebLab into a Web 2.0-enabled application which is also accessible from mobile devices. The structure of the paper is as follows. Section 2 offers an overview of the Web 2.0-compliant WebLab-Deusto system. Section 3 explains our progress transforming WebLab-Deusto into a mobile device accessible system. Finally, Section 4 draws some conclusions about our experiences and suggests further work.

2. A Web 2.0-enabled Remote Lab

WebLab-Deusto [2], [3] and [4] is a remote laboratory which allows the control of programmable PLD devices, specifically CPLDs and FPGA of Xilinx. Figure 1 and 2 show the Web front-end and the hardware made available for control at http://weblab.deusto.es.

A complete work session would consist of the following steps: The student writes a program in VHDL and generates the corresponding JEDEC file; the server receives the file and sends it to the CPLD board; the student controls the inputs of the application using the client software; the CPLD controls the outputs using the programmed algorithm; the student sees the changes through the webcam, which is constantly monitoring the CPLD and any attached devices and finally the user can try out other input signals, or close the connection with the server.

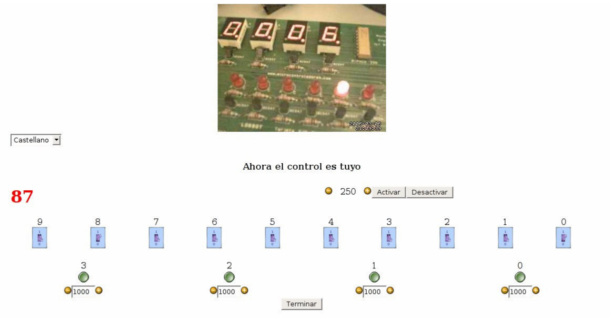


Figure 1. WebLab-PLD page

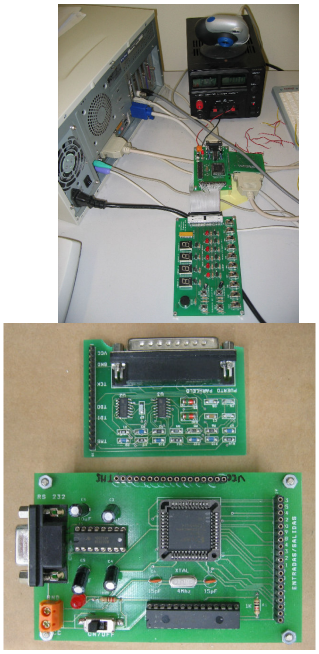


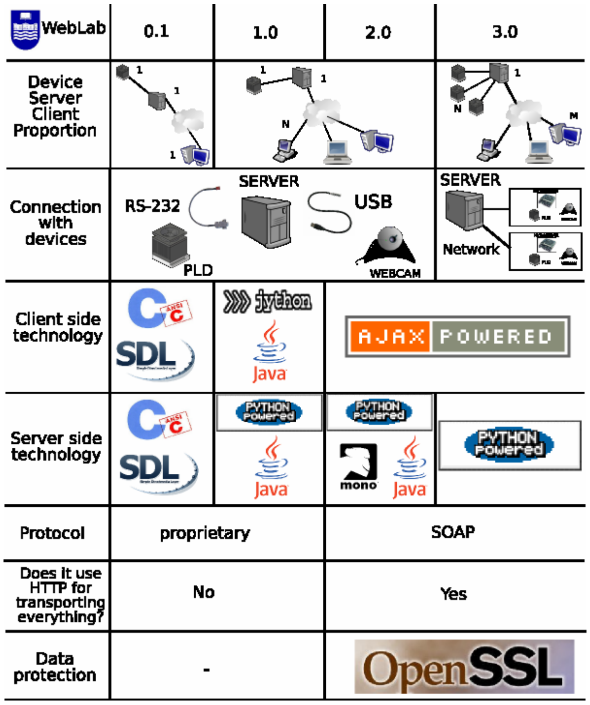
Figure 2. WebLab-PLD hardware

2.1. Architecture evolution of WebLab-Deusto

The architecture of the WebLab-Deusto has undergone several evolutions as shown in Fig.3.

The current stable architecture of WebLab-Deusto (v. 2.0) shown in Fig. 4 is an AJAX-based solution, i.e. a "Web 2.0-compliant" site. The server core has been developed in Python, while the wrapper to provide SOAP Web Services to clients has been implemented in Mono. Although version 2.0 and earlier ones depended on Microsoft Windows and Java, currently the system is transiting between 2.0 and 3.0 versions. In fact, WebLab-Deusto is now a cross-platform system running under GNU/Linux with no dependence on Java anymore.

AJAX is a new term used to refer to the successful combination of technologies such as XHTML and CSS for presentation, DOM, XML and XSLT for the model and JavaScript and its `XMLHttpRequest` object for the control, in order to build more interactive, dynamic and responsive web applications that resemble the behaviour (look & feel) of desktop applications. By means of AJAX a user can issue asynchronous calls to the server while he still remains in control of the web front-end. It is extensively and successfully used by well-known Web 2.0 applications, such as Gmail, Google Maps, or Flickr.



WebLab	0.1	1.0	2.0	3.0
Device Server Client Proportion	1 1 1	1 1 N		1 N M
Connection with devices	RS-232 SERVER USB PLD WEBCAM			SERVER Network
Client side technology	C ANSI C SDL	Jython Java	AJAX POWERED	
Server side technology	C ANSI C SDL	PYTHON powered Java	PYTHON powered mono Java	PYTHON powered
Protocol	proprietary		SOAP	
Does it use HTTP for transporting everything?	No		Yes	
Data protection	-		OpenSSL	

Figure 3. Evolution of WebLab-Deusto

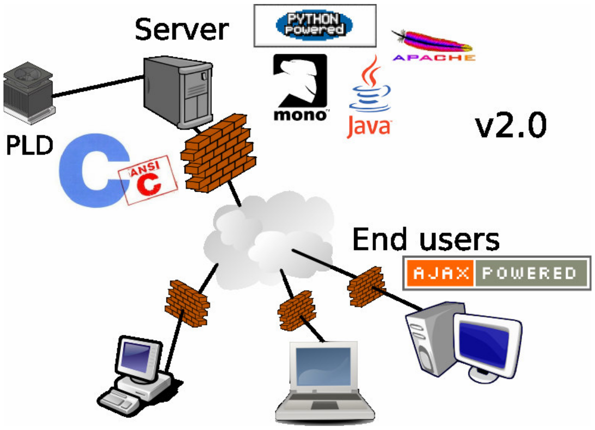


Figure 4. Architecture of WebLab-Deusto 2.0

Web front-end clients of WebLab-Deusto are run in a web browser, such as Microsoft Internet Explorer, Mozilla Firefox or Opera Web Browser. The communication between web browsers and server is based on SOAP, which offers a two-fold benefit: a) by means of AJAX a client can issue an asynchronous HTTP request to the server, making the whole user interaction more dynamic and flexible and b) facilitates the development of web-service compatible clients, not necessarily running on a Web browser, which can be run behind any kind of firewall or Web proxy.

3. A Mobility 2.0-enabled WebLab

Mobility 2.0 is trying to bring Web 2.0-enabled applications to the mobile domain. In a nutshell, it attempts to translate the benefits experienced by users of Web 2.0 applications on the desktop to mobile devices.

There are already some good examples of Mobility 2.0-enabled applications. For example, Google Local for Mobile (http://www.google.com/gmm/index.html) enables a user to access GoogleMaps from a Java enabled mobile running a J2ME application; Yahoo! Go Mobile (http://go.connect.yahoo.com/go/ mobile) is a J2ME client which enables a user access to Yahoo! Services such as Contacts, Email, Photos or Messenger from mobile devices; or even some Mobile Blog clients (Mobile Blog, Blogger) which populate blogs from mobile devices. All those applications are regarded as *mobile mash-ups*, i.e. they are web applications adapted to mobile devices combining content from several sources into an integrated experience. In order to develop such mobile mash-ups two main models have been followed:

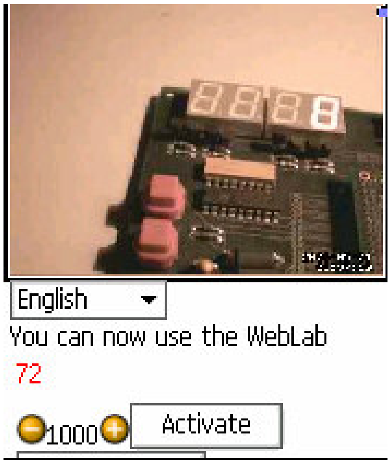
- *Browsing apps*, web apps which take into account limitations unique to mobility (e.g. small device or network bandwidth). The XHTML clients are capable of hardly any processing.
- *Smart Client apps*: downloaded and installed in the device. Clients are capable of some processing, storage and intermittent communication. Some example enabling technologies are J2ME, Compact.NET, Python for Series 60, BREW uiOne or Flash Lite

As we have already mentioned AJAX is a very important facet of Web 2.0. It avoids start-stop cycles emitting asynchronous calls to the server, so that the user does not wait. It solves two problems: a) superior UI experience and b) standardized form of data retrieval. Unfortunately, it does not have much presence on mobile devices. However, this situation is changing since many last generation mobiles come equipped with Opera Browser or Internet Explorer for Windows Mobile 5 which both support AJAX.

Following this emerging trend on the development of mobile applications we have been working on enabling our Web 2.0-enabled WebLab to be accessible from mobile devices, i.e. to turn into a Mobility 2.0 application.

As described in section 2, the client of the WebLab-Deusto is just a common web front-end running on a browser. It does not rely on any proprietary plug-in for the web browser such as Java Applets or Macromedia Flash. Thus, any Web Browser which implements the commonly used web standards required in AJAX is a potential client of the WebLab-Deusto. Due to this fact, it is easy to find clients running on different platforms, including Microsoft Windows, Mac OS, GNU/Linux or even mobile devices (Symbian, Windows Mobile).

The Opera web browser is a proprietary AJAX-compatible software available under many mobile platforms, like Nokia S60, S80, S90, SmartPhones with Windows Mobile, and so on. Any mobile device running the Opera Browser or the latest edition of Internet Explorer for Windows Mobile (also AJAX-compatible) can access the WebLab-Deusto (see Fig. 5), without changing anything in the architecture of the WebLab-Deusto. Currently, there are many ongoing projects aiming to develop both proprietary and open source AJAX-compatible web browsers for these devices. Therefore, it is feasible to assume that in the near future all mobile devices will be equipped with such browsers.



You can now use the WebLab

72

Figure 5: WebLab-FPGA from Opera Web Browser on a Nokia 6630

The other approach to access a WebLab from a cellular phone is programming a specific proprietary client for the mobile device. In order to avoid losing portability between different mobile devices, there are cross-platform development platforms available, mainly J2ME and Compact .NET. If WebLabs use a standard protocol for communication as is the case of WebLab-Deusto with SOAP, the development of clients with any of these technologies is simple. In fact, following this approach WebLab developers can, at the time being, aim at a wider range of devices.

The main drawback of the proprietary client approach is that it obviously requires the development and maintenance of a new client for any mobile platform supported. Moreover, every new feature in the WebLab should be ported to both the Web client and all the platform proprietary clients. With the AJAX approach followed in WebLab-Deusto, the user would automatically access the last version of the WebLab client every time she enters the WebLab's web site.

4. Conclusion and Further Work

Mobile devices offer ever increasing features (better screen, more natural interaction mechanisms, more sophisticated operating systems and development frameworks and so on), while communication networks offer better bandwidth and prices (there is a tendency towards cheaper data transmission costs and flat or even disappearing rates). These two arguments plus the intrinsic benefit of mobile devices, i.e. they are with us at anytime and anywhere, justify the use of mobile devices to undertake tasks we undertook before only from the desktop. One good example of this may be accessing WebLabs from mobile devices. Two of the potential benefits of enabling remote mobile access to WebLabs are:

- Students may work on their assignments while commuting from the university to their residences or while gathering for a meeting with other colleagues at a café or a park.
- Lab assignment supervisors may regularly, even when they are not close to a PC, monitor student work progress and act rapidly in case their intervention is required.

Having designed WebLab-Deusto around Web 2.0 principles has allowed us to seamlessly, effortlessly, enable access to it from mobile devices, without having to reinvent the wheel for any platform where a WebLab client may run. Nevertheless, mobile access to WebLab-Deusto is not universal. It is restricted to those devices equipped with an AJAX-capable browser, currently only restricted to top of the range mobile devices. However, it is foreseen that soon in the future all mobile devices will be equipped with more capable AJAX-compatible web browsers.

The application domain of the technology developed for WebLab-Deusto may be easily applied and adapted to other fields. A worker could, for example, reconfigure a manufacturing process with her mobile phone, or she could even modify the logic or software of the process, remotely, while on the move. Therefore, we plan to apply our experience on WebLabs to controlling and monitoring industrial machinery for preventive technical assistance purposes.

In order to foster WebLab adoption and acceptance, it is paramount to offer the most friendly, responsive and engaging web interface. We believe that adopting Web 2.0 design principles, as we have done in WebLab-Deusto, can answer those strict interaction requirements. Moreover, the presumed rapid evolution of mobile device browsers to adapt to Web 2.0 application technical requirements, will allow anybody to access WebLab-Deusto from mobile devices, still having a very satisfactory user-experience.

Acknowledgments

This work has kindly been supported by the “Cátedra de Telefónica Móviles” at the Faculty of Engineering in the University of Deusto.

References

[1] O'Reilly T. *What Is Web 2.0*, September 2005, http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[2] García-Zubía, J.; López-Ipiña, D.; Orduña, P. “Towards a canonical software architecture for multidevice weblabs", *31st Annual Conference of the IEEE Industrial Electronics Society, IECON 2005*, ISBN: 0-7803-9253-1, pp: 2146-2151, North Carolina (EEUU).

[3] García-Zubía, J.; Moral, A.López-Ipiña, D.; Orduña, P. “Suitability and implementation of a WebLab in engineering", *10th International Conference on Emerging Technologies and Factory Automation, ETFA 2005*, ISBN: 0-7803-9402-X, Vol II, pp: 49-56, Catania (Italia).

[4] García-Zubía, J.; López-Ipiña, D.; Orduña, P. *Evolving towards better architectures for remote laboratories: a practical case*, International Journal of Online Engineering, Special Issue REV 2005, http://www.ijoe.org/ojs/viewissue.php?id=2